Taking a Quantum Leap in Time to Solution for Simulations of High-Tc Superconductors

Peter Staar
ITP, ETH Zurich
staarp@itp.phys.ethz.ch

Thomas A. Maier
CSMD, Oak Ridge Natl. Lab.
maierta@ornl.gov

Raffaele Solca
ITP, ETH Zurich
rasolca@itp.phys.ethz.ch

Gilles Fourestey
CSCS, ETH Zurich
fourestey@cscs.ch

Michael S. Summers
CSMD, Oak Ridge Natl. Lab.
x7u@ornl.gov

Thomas C. Schulthess
ITP & CSCS, ETH Zurich
CSMD, Oak Ridge Natl. Lab.
schulthess@cscs.ch

ABSTRACT

We present a new quantum cluster algorithm to simulate models of high-Tc superconductors. This algorithm extends current methods with continuous lattice self-energies, thereby removing artificial long-range correlations. This cures the fermionic sign problem in the underlying quantum Monte Carlo solver for large clusters and realistic values of the Coulomb interaction in the entire temperature range of interest. We find that the new algorithm improves time-to-solution by nine orders of magnitude compared to current, state of the art quantum cluster simulations. An efficient implementation is given, which ports to multi-core as well as hybrid CPU-GPU systems. Running on 18,600 nodes on ORNL's Titan supercomputer enables us to compute a converged value of $T_c/t = 0.053 \pm 0.0014$ for a 28 site cluster in the 2D Hubbard model with $U/t = 7$ at 10% hole doping. Typical simulations on Titan sustain between 9.2 and 15.4 petaflops (double precision measured over full run), depending on configuration and parameters used.

1. INTRODUCTION

For over a century, superconductivity has fascinated generations of physicists and materials scientists, and continues to captivate researchers as new classes of superconducting materials are discovered. Below a certain critical temperature T_c, superconductors exhibit conductance without resistance as well as perfect diamagnetism, i.e. a complete expulsion of an external magnetic field (Meissner-Ochsenfeld effect), implying that the electrons behave in a collective manner. This is possible, because of a net attractive interaction forcing electrons to form boson-like Cooper pairs, which can condense into a coherent macroscopic quantum state analogous to a Bose-Einstein condensate. The energy required to break up pairs (also called energy gap) suppresses the scattering processes from defects and impurities that would give rise to electrical resistance in normal conductors.

In conventional superconductors, which include many elemental metals such as Hg, Al and Nb, the attractive interaction between electrons arises from the interaction between the negatively charged electrons and the positively charged ions. The distortion of the ion lattice left behind by the motion of an electron attracts a second electron and thus results in an effective attractive interaction between the electrons. This mechanism is active and can overcome the instantaneous Coulomb repulsion at long times, or low energies, because the ion dynamics is slow compared to the electrons. This retardation, however, also limits the magnitude of T_c that can be reached in these systems. These concepts are well described and understood within a rigorous theoretical foundation, the BCS (Bardeen-Cooper-Schrieffer) theory [6], and its extension, the Migdal-Eliashberg theory [23, 7].

The high-temperature superconducting copper-oxide based materials (high-T_c cuprates), discovered in 1986, have much higher critical temperatures, up to 150 K, exceeding by far the theoretical limit of conventional superconductors. Early on this triggered speculations that superconductivity in these systems arises from a different mechanism. In addition, the spatial component of the pair wave function was found to have $d_{x^2-y^2}$-wave structure, different from the isotropic s-wave structure in conventional superconductors. It is by now generally accepted that pairing in these systems has a different origin and most likely arises from the strong magnetic interactions or fluctuations between the electron spins that result in an antiferromagnetic phase in the undoped parent compounds [25]. The details of how superconductivity emerges from the highly enigmatic pseudogap phase, which dominates much of the higher temperature phase diagram of the doped systems, is still under heavy debate.

The physics of these crystalline systems is well described by a Hamiltonian that is expanded in atomic orbitals

$$\mathcal{H} = \sum_{ij,\mu\nu,\sigma} t_{ij}^{\mu\nu} c_{i\mu\sigma}^{\dagger} c_{j\nu\sigma} + \sum_{ij,\mu\nu,\sigma\sigma'} U_{ij}^{\mu\nu,\sigma\sigma'} n_{i\mu\sigma} n_{j\nu\sigma'}, \quad (1)$$

where the fermionic creation (destruction) $c_{i\mu\sigma}^{\dagger}$ ($c_{i\mu\sigma}$) operator creates (destroys) an electron with spin σ in atomic orbital μ on lattice site i, $n_{i\mu\sigma} = c_{i\mu\sigma}^{\dagger} c_{i\mu\sigma}$ is the corresponding number operator and non-density-density interactions have been neglected. The transition amplitudes $t_{ij}^{\mu\nu}$ between orbitals on different sites and Coulomb interactions $U_{ij}^{\mu\nu,\sigma\sigma'}$ can be computed from ab initio electronic structure calcula-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SC13 November 17-21, 2013, Denver, Colorado, USA
Copyright 2013 ACM 978-1-4503-2378-9/13/11
http://dx.doi.org/10.1145/2503210.2503282 ...$15.00.

tions [2, 3, 4, 5, 15, 22, 24] – these quantities decay rather fast with distance between lattice sites due to the spatial locality of the atomic orbitals and screening effects.

However, despite the simplicity of the Hamiltonian, the theoretical description of these systems is extremely challenging. Traditional many-body perturbation theory cannot be used, due to the absence of a small parameter, and current implementations based on density functional theory (DFT), are unable to adequately describe the strong electronic correlations that underly the physics of this system, particularly the physics of the superconducting state. However, DFT calculations of the electronic ground state show – in remarkable agreement with photoemission experiments – that the low-energy electronic structure is governed by a single band of electrons formed by the Cu $3d_{x^2-y^2}$ and the O $2p_{x/y}$ orbitals of the CuO planes. The physics of this band may be described by a single-band Hubbard model

$$\mathcal{H} = -t \sum_{\langle ij \rangle, \sigma} c^{\dagger}_{i\sigma} c_{j\sigma} + U \sum_{i} n_{i\uparrow} n_{i\downarrow} \,, \qquad (2)$$

where the sums run over the sites i of a 2D square lattice and the notation $\langle ij \rangle$ indicates that the j-sumation runs only over the four near neighbor sites of i. The second term describes an on-site Coulomb repulsion that raises the energy when two electrons reside on the same site and therefore favors singly occupied sites and thus moment formation. Numerical calculations based on this model have found many phenomena similar to what has been observed experimentally, including antiferromagnetism, pseudogap behavior and strong d-wave pair correlations, and are thought to be the simplest to capture qualitatively the relevant physics of superconductivity in the cuprates. The reliable computation of the superconducting transition temperature, however, has thus far posed insurmountable computational challenges for even this simplified model.

Simulations based on the Dynamical Cluster Approximation (DCA) showed that the model describes a superconducting transition with d-wave character [19], and even allowed an analysis of the pairing mechanism [20]. But they were far from converged due to a notorious cluster shape dependence of the results computed within the DCA. Furthermore, the simulations of T_c were done at an unrealistically small value of $U/t = 4$, since the fermionic sign problem of the quantum Monte Carlo algorithms used as a cluster solver within the DCA, prevented simulations for values of $U/t = 7$ that would be more realistic for these systems.

A quantitative description that accounts for, e.g., materials specific variation of T_c between different cuprates, requires calculations based on the more complex multi-orbital model, see Eq. (1), which explicitly take into account additional bands and orbitals and thus can distinguish between different materials. Extending the computation to multiple orbitals and inclusion of realistic, materials specific electronic structure is not conceptually difficult. However, plans to do so have remained elusive, as the reliable solution of even the simple single-band Hubbard model has posed insurmountable computational challenges.

In the present paper we discuss a new algorithm and its implementation on multi-peta flops supercomputers, which for the first time allows us to compute reliably the superconducting transition temperature of the single-band 2D Hubbard model with realistic values for the on-site Coulomb interaction, i.e. $U/t = 7$. The state of the art in quantum cluster simulations is presented in section 2. In section 3 we discuss the new DCA$^+$ algorithm, as well as its impact on reducing the QMC sign problem. Implementation and details of performance measurements for multi-core and hybrid CPU-GPU systems are given in section 4. The computer systems used in this study as well as a performance model of the simulation code are presented in section 5. This discussion includes a comparison between multi-core and hybrid CPU-GPU systems that is based on both time and energy to solution. In section 6 we analyze the performance of our simulations at scale. The analysis shows a dramatic nine-order-of-magnitude improvement in time to solution compared to the sate of the art that is due to the reduction of the QMC sign problem, and it shows that strong and weak scaling behavior are well understood in terms of the performance model - sustained performance when running typical full production simulation on 18,600 nodes of the Titan supercomputer at ORNL was between 9.2 and 15.4 (double precision) petaflops, depending on the problem configuration. We conclude the paper with a summary and a discussion of future work in section 7.

2. THE STATE OF THE ART IN QUANTUM CLUSTER SIMULATIONS

The challenge for simulations of quantum many-body problems, put in simple terms, is that the complexity increases exponentially with system size. Specifically, the dimensionality of the Hilbert space of the Hamiltonian operators in equations (1) and (2) increases exponentially in the number of sites and orbitals. Describing the superconducting phase, as any other symmetry broken state, generally requires a simulation of a Hubbard model in the thermodynamic limit, with a number of sites $N \rightarrow \infty$. This is clearly intractable for any classical computer. Therefore, this type of problems in materials science is approached with effective medium theories [8] and, more specifically, dynamical mean-field theory (DMFT) [14] for Hubbard model type of problems in condensed matter physics. For the study of superconductivity, where non-local correlations have to be treated explicitly, DMFT has to be extended with quantum cluster approaches [18]. These techniques replace the bulk lattice problem by a finite size cluster embedded in a dynamic mean-field host, which is designed to describe the remaining degrees of freedom. Non-local dynamical correlations are thus treated accurately up to a length-scale set by the cluster size, while the effects of longer-ranged correlations are approximated by the mean-field. The dynamical cluster approximation (DCA) is a quantum cluster method that has been used successfully to study the superconducting transition in Hubbard-type models [11, 13]. In the DCA, the macroscopic lattice problem is reduced to a finite site quantum cluster impurity with periodic boundary conditions, embedded self-consistently into a mean-field. This reduction is achieved via a coarse-graining procedure of the Green's function $G(\vec{k}, \varpi_m)$, in which the domain of $\vec{k}$, the Brillouin zone, is divided into N_c patches and the self-energy $\Sigma(\vec{k}, \varpi_m)$ is assumed to be constant on these patches. For the purpose of this discussion, Σ is defined through Dyson's equation $G_0^{-1} - G^{-1} = \Sigma$, where G_0 is the Green's function corresponding to the non-interacting system. Formally, the

self-energy is thus parametrized as

$$\Sigma(\vec{k}, \varpi_m) = \sum_i \phi_{\vec{K}_i}(\vec{k})\, \Sigma_{\vec{K}_i}(\varpi_m), \tag{3}$$

where the set of patches $\{\phi_{\vec{K}_i}(\vec{k})\}$ is defined through the cluster-momenta $\{\vec{K}_i\}$,

$$\phi_{\vec{K}_i}(\vec{k}) = \begin{cases} 1 & \forall j : |\vec{k} - \vec{K}_i| \leq |\vec{k} - \vec{K}_j| \\ 0 & \exists j : |\vec{k} - \vec{K}_i| > |\vec{k} - \vec{K}_j| \end{cases} \tag{4}$$

and $\Sigma_{\vec{K}_i}(\varpi_m)$ is the cluster self-energy for momentum $\vec{K}_i$. For notational convenience, we will drop the Matsubara frequencies ϖ_m from hereon. The DCA method is depicted

Algorithm 1 The DCA algorithm

1: Start with a cluster self-energy $\Sigma_{\vec{K}}$.
2: **repeat**
3: Adjust the chemical potential μ
4: coarse-grain the Greens function $\bar{G}_{\vec{K}} = \int d\vec{k}\, \phi(\vec{k} - \vec{K})\, G(\vec{k})$
5: Compute the cluster-excluded Greens functions $G^0_{\vec{K}} = \left[\bar{G}^{-1}_{\vec{K}} + \Sigma_{\vec{K}}\right]^{-1}$
6: Solve the cluster-impurity problem and obtain a new cluster self-energy $\Sigma_{\vec{K}}$.
7: **until** $\Sigma_{\vec{K}}$ is converged

in Algorithm 1. One starts the calculations with an initial guess for the self-energy, which may simply be zero for all values of the vector $\Sigma_{\vec{K}}$, and iterates the DCA loop to convergence. The essential steps in a DCA iteration are: coarse-graining the Green's function; computing the cluster excluded Green's function, which in the DCA represents the effective medium, by subtracting the cluster self-energy; and solving the cluster impurity problem to obtain a new cluster self-energy. The hard problem that needs to be solved in the DCA is the numerical solution of the cluster impurity problem. The most accurate and versatile method for this so-called cluster solver is Quantum Monte Carlo (QMC) integration. The state of the art algorithm in 2008, when some of us ran the first sustained peta-flops simulations on the Cray XT5, Jaguar at ORNL [1] was the Hirsch-Fye QMC (HF) algorithm [12] with delayed updates [1]. Since then, a new class of algorithms that remedy the time-discretization problem of the HF method have emerged. In particular, the continuous time auxiliary field QMC algorithm (CT-AUX) [10] has been developed specifically for the type of impurity problems that has to be solved in the models of high temperature superconductivity. The CT-AUX has been improved in two ways: (1) a submatrix update algorithm has been implemented that, similarly to the delayed updates for the HF algorithm increases data locality and thus computational efficiency on modern processors [26, 9] and (2) accumulation of measurements with non-equidistance fast Fourier transforms that reduced the complexity with controlled accuracy [30]. In principle, the complexity of the QMC cluster solver scales with $\mathcal{O}(U^3 N_c^3/T^3)$ for a fixed number of measurements and the DCA algorithm will reproduce the exact solution of the lattice model in the limit of infinite cluster size. In practice, however, the fermionic sign problem [16] leads to an exponential increase in measurements required to reach a given accuracy for clusters above a certain size. In the single-band Hubbard model with a realistic $U/t = 7$, the largest cluster for which T_c can be calculated has only 10 sites. For a smaller Coulomb interaction of $U/t = 4$, clusters as large at 26 could be reached. Unfortunately, neither the algorithmic improvements (CT-AUX, sub-matrix updates, and non-equidistant FFT kernels), nor the order of magnitude increase in peak performance of the largest supercomputers since 2008 are significant enough to beat the exponential decay of the fermionic sign. The time to solution will explode exponentially when one attempts to further increase the cluster size. Given the upper limit in cluster size that can be reached in the simulations, the most apparent weakness of the DCA algorithm is the cluster shape dependence and related poor convergence of the results. This is illustrated with the red dots in figure 1, where the computed values for the superconducting transition temperature T_c as determined with the DCA for a Hubbard model with $U/t = 4$ are shown as a function of the cluster size N_c. Not only are the results scattered for the different cluster sizes, they show a strong dependence on the cluster shape. For example the results for two different 16-site clusters shown in the figure differ by 40%. This feature of the DCA is directly related to the geometric shape dependence of the coarse graining patches of equation (4), which carries over into the lattice self-energy via equation (3).

3. A NEW ALGORITHM: DCA$^+$

The key to fixing the convergence problem of T_c versus cluster size in the DCA is to extend the algorithm in such a way that the lattice self-energy is continuous and independent of the shape of the cluster. All previous attempts to accomplish this were based on interpolation of the cluster self-energy $\Sigma_{\vec{K}}$ and were unsuccessful, as they introduced causality problems [27]. Here we introduce an algorithm, in which the interpolation of the cluster self-energy and the inference of the lattice self-energy are separated in distinct steps. Since the new algorithm can be viewed as an extension of the DCA with a continuous lattice self-energy, we call it DCA$^+$ [29]. The new algorithm, which we discuss in subsection 3.1 shows much better convergence properties as can be seen from the black triangels in Fig. 1. Its main impact, however, is that it significantly delays the fermionic sign problem of the QMC, such that simulations at much larger clusters are now possible. Not only can the results for $U/t = 4$ be converged (see Fig. 1), but simulations with the realistic values for the Coulomb interaction $U/t = 7$ become feasible for large clusters as well. Thus, with the DCA$^+$ algorithm and its implementation on multi peta-flops supercomputers that will be discussed in the subsequent sections of this paper, we are now in a position to perform converged quantum cluster simulations and determine the superconducting transition temperature for realistic models of the cuprates and other high temperature superconductors.

3.1 The algorithm

By multiplying Eq. (3) on both sides with a coarse-graining patch $\phi_{\vec{K}'}(\vec{k})$ and integrating over the Brillioun zone, the DCA$^+$ algorithm emerges from the DCA in a natural way. Using the identity $\int d\vec{k}\, \phi_{\vec{K}}(\vec{k})\, \phi_{\vec{K}'}(\vec{k}) = \delta_{\vec{K},\vec{K}'}$, we have effectively inverted Eq. (3) and obtain that the cluster self-energy $\Sigma_{\vec{K}}$, i.e. the self-energy obtained from the cluster-solver, should equal the coarse-grained lattice self-energy $\bar{\Sigma}_{\vec{K}}$

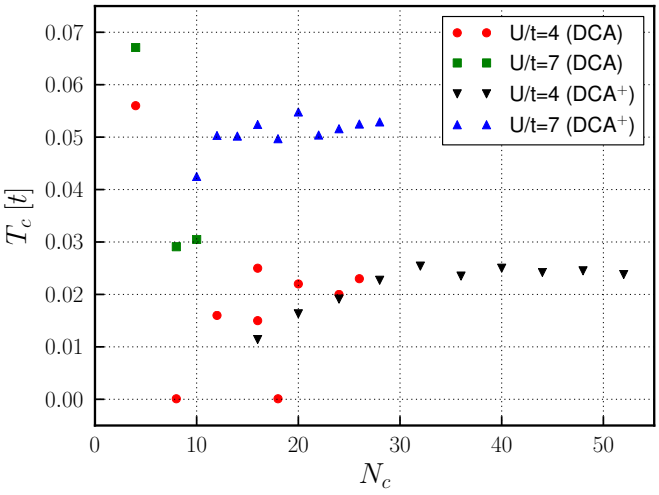

Figure 1: T_c versus cluster-size N_c in the weakly ($U/t = 4$) and strongly ($U/t = 7$) interacting regime. T_c is determined by locating the temperature at which the pairfield susceptibility diverges. For a realistic hopping-parameter of $t \approx 0.25\,eV$, we obtain a T_c of respectively 125K and 75K in the strong and weak interacting regime.

$$\Sigma_{\vec{K}_i} = \int d\vec{k}\, \phi_{\vec{0}}(\vec{k} - \vec{K}_i)\, \Sigma(\vec{k}) \equiv \bar{\Sigma}_{\vec{K}_i} \tag{5}$$

Eq. (5) forms the basic equation of the DCA$^+$ and is fundamentally different from Eq. (3) in that it allows the lattice self-energy $\Sigma(\vec{k})$ to be continuous. Together with the coarse-graining of the Greens-function, Eq. (5) tells us how to map the infinite lattice onto a finite cluster. It also dictates the convergence criterium of the DCA$^+$, namely that the coarse-grained lattice self-energy needs to be equal to the cluster self-energy. In practice, this condition is relaxed since the cluster self-energy is only known up to some statistical noise ϵ_{QMC}. Consequently, convergence is reached when the difference between the coarse-grained lattice self-energy $\bar{\Sigma}_{\vec{K}}$ and cluster self-energy $\Sigma_{\vec{K}}$ is less than ϵ_{QMC}.

The non-trivial part of the DCA$^+$ consists of generating in a numerically stable way the lattice-self-energy, given a cluster self-energy. In a simplistic approach, one could reinterpret the coarse-graining in Eq. (5) as a convolution. Using the convolution-theorem, we could therefore obtain the real-space Fourier components of the lattice-self-energy, which after a Fourier transform would give us the lattice self-energy,

$$\Sigma(\vec{k}) = \sum_{\vec{R}} e^{-\imath\, \vec{k}\, \vec{R}} \Sigma_{\vec{R}} / \phi_{\vec{R}} \tag{6}$$

This approach, however, is numericall problematic for two reasons. First, the sum over cluster-sites is finite and thus introduces a sharp cut-off. This in turn leads to causality problems in the self-energy due to the Gibbs phenomenon [27]. Another problematic issue with this simple approach is the fast decay of the real space components of the coarse-grain patch $\phi_{\vec{R}}$ at large cluster-radius $|\vec{R}| \gg 1$. The DCA relies on the fact that the self-energy is a smooth function in momentum-space, and thus a localized function in real space. The rapidly decaying real space components $\phi_{\vec{R}}$ might prevent this, by blowing up real space components of the self-energy far from the origin.

For these reasons, we have proposed to do the lattice-mapping in two steps [29]. We first want to do a controlled interpolation of the cluster self-energy from the discrete cluster-momenta $\vec{K}$ to arbitrary k-points in momentum space. Next, we perform a deconvolution on the interpolated self-energy $\tilde{\Sigma}(\vec{k})$, using a Bayesian inference scheme commonly used in image restoration.

To do a controlled interpolation, we propose to first apply an injective, analytical transformation $\mathcal{T}(z) = (z - \imath)^{-1}$ on the self-energy. For positive matsubara frequencies, the imaginary part of the cluster self-energy is less than zero and the transformed self-energy $\mathcal{T}(\Sigma_{\vec{K}})$ will become much smoother and we can safely apply the Wannier-interpolation technique [21]

$$\tilde{\Sigma}(\vec{k}) = \mathcal{T}^{-1}\Big[\sum_{\vec{R}} e^{-\imath\, \vec{k}\, \vec{R}} \underbrace{\Big(\sum_{\vec{K}} e^{-\imath\, \vec{k}\, \vec{R}} \mathcal{T}(\Sigma_{\vec{K}})\Big)}_{(\mathcal{T}\Sigma)_{\vec{R}}}\Big]. \tag{7}$$

This interpolation procedure is a controlled operation, since we verify whether $(\mathcal{T}\Sigma)_{\vec{R}}$ converges on the cluster. Another important aspect of this procedure is that it does not introduce any *extra* features in the interpolated self-energy, if the convergence criterium is satisfied. This is typically the case with splines, where overshoots aren't uncommon. Once we have interpolated the cluster self-energy, we can symmetrize the interpolated self-energy $\tilde{\Sigma}(\vec{k})$ according to the lattice symmetry-group. This symmetrization is also a tell-tale mark of the DCA$^+$, which cures the cluster shape dependence of the results. The strict parametrization of the lattice self-energy in the DCA does not allow the lattice self-energy to be symmetrized, unless the cluster symmetry group is equal to the lattice symmetry group.

To obtain the lattice self-energy, we generalize Eq. (5) by replacing the cluster self-energy with the interpolated cluster self-energy,

$$\tilde{\Sigma}(\vec{k}') = \int d\vec{k}\, \phi_{\vec{0}}(\vec{k} - \vec{k}')\, \Sigma(\vec{k}) \tag{8}$$

Since the patch $\phi_{\vec{0}}(\vec{k})$ acts as a box-car filter on the lattice self-energy, we can apply the Richardson-Lucy algorithm [17, 28] to solve this deconvolution problem. The Richardson-Lucy algorithm is a Bayesian inference scheme, which finds the most likely solution in an iterative way

$$\Sigma_{i+1}(\vec{k}) \leftarrow \Sigma_i(\vec{k}) \int d\vec{k}'\, \frac{\phi_0(\vec{k} - \vec{k}')\, \tilde{\Sigma}(\vec{k}')}{\int d\vec{k}''\, \phi_0(\vec{k}' - \vec{k}'')\, \Sigma_i(\vec{k}'')}. \tag{9}$$

In algorithm 2, we have summarized the DCA$^+$ algorithm. We generally start from a continuous self-energy, either zero or obtained from second order perturbation theory. Next, we adjust the chemical potential to satisfy our desired density. The lattice is then mapped on the cluster by coarse-graining the Greens-function and the self-energy. A cluster excluded Greens function is generated from the coarse-grained Greens-function and self-energy and plugged into the cluster-solver. The latter produces a new cluster self-

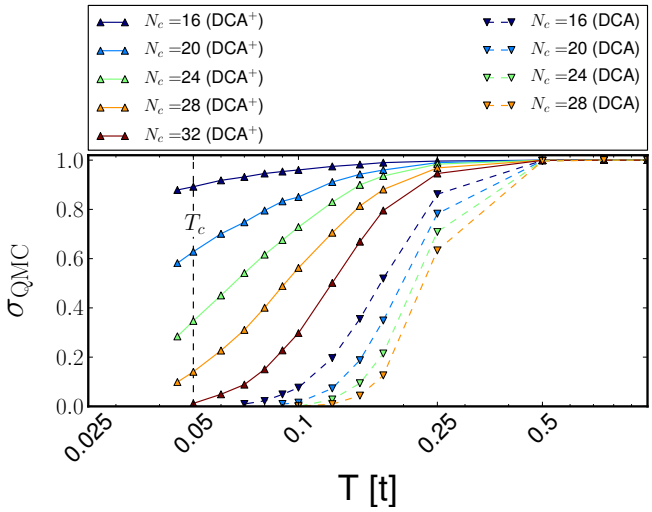

Figure 2: The evolution of the fermionic sign with temperature for the DCA and DCA$^+$.

energy, which after a successive interpolation, symmetrization and deconvolution generates a new lattice self-energy. This process is repeated until the cluster self-energy and coarse-grained lattice self-energy are converged up to the Monte Carlo error ϵ_{QMC}.

Algorithm 2 DCA$^+$

1: Start with a continuous self-energy $\Sigma(\vec{k})$.
2: **repeat**
3: Adjust the chemical potential μ
4: coarse-grain the self-energy and Greens function
$\bar{G}_{\vec{K}} = \int d\vec{k}\, \phi(\vec{k}-\vec{K})\, G(\vec{k})$,
$\bar{\Sigma}_{\vec{K}} = \int d\vec{k}\, \phi(\vec{k}-\vec{K})\, \Sigma(\vec{k})$
5: Compute the cluster-excluded Greens functions
$G^0_{\vec{K}} = \left[\bar{G}^{-1}_{\vec{K}} + \bar{\Sigma}_{\vec{K}}\right]^{-1}$
6: Solve the cluster-impurity problem and obtain a new cluster self-energy $\Sigma_{\vec{K}}$.
7: Interpolate the self-energy $\{\vec{K}, \Sigma_{\vec{K}}\} \rightarrow \tilde{\Sigma}(\vec{k})$
8: Symmetrize the interpolated self-energy $\tilde{\Sigma}(\vec{k})$ according to the symmetry group of the lattice.
9: Invert the generalized coarse-graining
$\tilde{\Sigma}(\vec{k}') = \int d\vec{k}\, \phi(\vec{k}-\vec{k}')\, \Sigma(\vec{k})$,
using the Richardson-Lucy deconvolution algorithm.
10: **until** $|\bar{\Sigma}_{\vec{K}} - \Sigma_{\vec{K}}| < \epsilon_{QMC}$

3.2 Reduction of the QMC sign problem

The main benefit of introducing a continuous lattice self-energy, as it turns out, is a significant delay of the QMC sign problem that dramatically reduces the time to solution for large N_c, allowing us to reach convergence in the computation of the superconducting transition temperature T_c with extreme-scale, multi-petaflop supercomputers, as is demonstrated in Fig. 1.

It has already been demonstrated [13] that the inclusion of the effective medium reduced the QMC sign problem in the DCA compared to regular finite system size QMC calculations. The introduction of a continuous lattice self-energy in the DCA$^+$ further reduces the sign problem as can be understood intuitively in the following way. As a consequence of the discontinuities in momentum space, the DCA self-energy has Fourier coefficients that decay in real space with $1/|\vec{R}|$. On the other hand, the continuous lattice self-energy of the DCA$^+$ has real-space Fourier coefficients that decay faster, proportionally to $1/|\vec{R}|^2$. The DCA$^+$ thus removes artificial, long-range correlations of the DCA method. Empirically, it is known that the fermion sign problem worsens with increasing strength of correlations. Therefore we believe that the removal of artificial long-range correlations reduces the sign problem in the DCA$^+$ compared to the DCA. This is illustrated in Fig. 6, where we plot the measured QMC sign in DCA and DCA$^+$ calculations, respectively, for various cluster sizes N_c as a function of temperature. It can be seen, that the QMC sign remains intact (i.e. above 0.5) in the DCA$^+$ for clusters as large at $N_c = 20$ with realistic Coulomb interaction of $U/t = 7$. Even in the case of large cluster such as $N_c = 28$, the QMC sign is still large enough that temperatures below T_c can be reached with very large simulations (see figure 1 and section 6).

4. IMPLEMENTATION AND METHOD OF PERFORMANCE MEASUREMENTS

Just as in state of the art DCA, the heart of the DCA$^+$ simulations is the CT-AUX algorithm [10] with submatrix updates [9] and accumulation of measurements with non-equidistant fast Fourier transforms [30]. In the CT-AUX methodology, the partition function is expressed as an expansion in terms of N-matrices that are related to the Green's function through $G = N\, G_0$,

$$\mathcal{Z} = \sum_{k\geq 0} T^k \sum_{\{(\tau_i,s_i)\}_k} \prod_{\sigma=\uparrow,\downarrow} |N^\sigma_{\{\tau_i,s_i\}_k}|^{-1}, \qquad (10)$$

where the inner sum runs over k auxiliary spin fields s_i are taken at arbitrary imaginary times τ_i. Following the Metropolis algorithm, the partition function is sampled by randomly creating and annihilating auxiliary spins at different times with different orientation. A Dyson-like equation relates the N-matrices between two different auxiliary field configurations. In the case of a single spin update, this takes the form of a rank-1 update for N^σ, which is sketched in Fig. 3A. With a blocking technique commonly used in the LU-decomposition, k_s subsequent rank-1 updates can be rewritten as a rank-k_s update [26, 9], which schematically is depicted in Fig. 3B. Here, k_s is a tuning parameter that is adjusted to maximize computational efficiency, since multiple vector outer products (DGER) are bundled into a single matrix-matrix multiplication (DGEMM).

The usual way to parallelize Monte Carlo algorithms is to assign a Markov chain to every MPI rank, and one rank per compute core of a parallel computer. Measurements are performed in order with the Monte Carlo updates and accumulated on every core. Communication between ranks is limited to a reduction operation at the end of the Monte Carlo simulation – note that in the present case, the quantum Monte Carlo simulation is just one step inside the respective DCA and DCA$^+$ iterations (see Algorithms 1 and 2). In light of modern machines with hybrid nodes consisting of multi-core CPU and GPU, we use a different implementation strategy. We run one MPI rank for each node and use $N_w + N_a$ pthreads per node, where N_w and N_a

are, respectively, the numbers of walker and accumulator threads running on each node. On a multi-socket CPU node, the walker and accumulator threads are distributed evenly over the processors. On hybrid nodes, the compute intensive walker threads run on the GPU, since they consist mostly of matrix multiplications (see Fig. 3). The accumulators that execute measurements, which typically are more complex operations that benefit from latency optimized cores, run on the multi-core CPU. When a simulation starts up, all walkers begin from a random configuration running updates until the average expansion order $\langle k \rangle$ in eq. (10) converges, at which point the particular walker is thermalized. At this point the stochastic sampling of the Green's function and other observables begins. Walkers send their data (the N-matrix) over to the pool of accumulator threads, where the measurements are processed asynchronously. On a hybrid system this involves transferring data from the GPU to the CPU over the PCIe bus. Should all accumulators be busy, the walker continues to update its configuration. However, given the speed of the measurements with non-equidistant fast Fourier transforms, the accumulators are typically ready to take on measurements before the walkers are done with a minimum sweep of $\langle k \rangle$ updates required between measurements. This implementation as it runs on a hybrid CPU-GPU node is summarized in algorithm 3.

On the CPU only version of the code, the floating point performance is measured by counting all floating point operations with PAPI counters and dividing by the entire wall-clock time of the *entire* simulation. Since such counters are not available on GPUs, a lower bound for the floating point operations performed when running simulations on a hybrid CPU-GPU system is computed analytically from the matrix dimensions used on the sub-matrix updates of the CT-AUX algorithm. Three DGEMM and one DTRSM operation are necessary for a submatrix update in each of the two spin channels. Two of these DGEMM are initialization steps. The actual update, represented in Fig. 3B, involves one DGEMM and one DTRSM. Each DGEMM will contribute $2\,k_s\,\langle k \rangle^2$ FLOPs, while the DTRSM accounts for

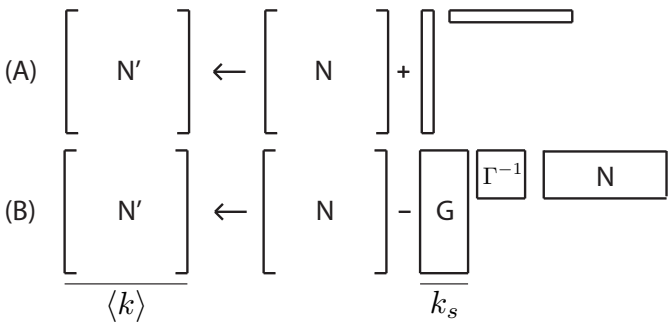

Figure 3: (A) Sketch of a rank 1 update with DGER: The column and row are respectively the column of Greens-function matrix G and the row of N, corresponding to the updated spin. (B) Sketch of a submatrix-update with DGEMM: The fat column and fat row are the bundled columns of G and the bundled rows of N, corresponding to the updated spins. The Γ-matrix is the Greens-function matrix connecting all updated spins.

Algorithm 3 Hybrid-multicore implementation

```
Start with N_w walkers and N_a accumulators.
while walker is not thermalized do
   1 submatrix-update (GPU)
end while
repeat
   if walker then
      do ⟨k⟩/k_s submatrix-updates (GPU)
      while all accumulators are busy do
         1 submatrix-update (GPU)
      end while
   end if
   copy configuration from the walker (GPU) to the accumulator (CPU)
   if accumulator then
      do measurement (NFFT on CPU)
   end if
until all accumulators are finished
```

$k_s^2\,\langle k \rangle$ FLOPs. Comparing this lower bound estimate with identical simulations running on CPU-only system where we measure all floating point operations, we observe that the discrepancy is never larger than 5%.

5. COMPUTER SYSTEMS AND PERFORMANCE MODEL

We use a single code base for the implementation of the DCA and DCA$^+$ algorithms (outer loop see algorithm 1 and 2), as well as the quantum Monte Carlo cluster solver described in the previous section, on distributed multi-core and hybrid CPU-GPU systems. The code is implemented in `C++` with template meta-programming techniques to hide architectural complexities in the quantum Monte Carlo implementation. We use MPI for inter-node parallelization, pthreads for the multi-threading on the nodes, and CUDA for the hybrid implementation. The simulations reported in this work have been executed on distributed multi-core and distributed hybrid CPU-GPU architectures. The following computer systems are used for the performance results we report in this and the next section:

Monte Rosa is a Cray XE6 distributed multi-core system at the Swiss National Supercomputing Center (CSCS) consisting of 1,496 compute nodes with two AMD Interlagos processors running at 2.1 GHz and 16 cores each. Each node is equipped with 32 GB DDR3-1600 RAM.

Todi is a Cray XK7 distributed hybrid CPU-GPU system at CSCS consisting of 272 compute nodes with one 16 core AMD Interlagos processor running at 2.1 GHz, and one NVIDIA K20X GPU that has 14 streaming multi-processors running at 732 MHz. The compute nodes are equipped with 32 GB DDR3-1600 memory and the GPU has 6 GB GDDR5 memory.

Titan is a Cray XK7 distributed hybrid CPU-GPU system at ORNL consisting of 18688 compute nodes with one 16 core AMD Interlagos processor running at 2.2 GHz, and one NVIDIA K20X GPU that has 14 streaming multi-processors running at 732 MHz. The compute nodes are equipped with 32 GB DDR3-1600 memory and the GPU has 6 GB GDDR5 memory.

Practically all the time of a fully self-consistent DCA$^{(+)}$ calculation is spend in the CT-AUX cluster solver. In or-

der to understand the performance of the code on different architectures, it is thus sufficient to analyze the quantum Monte Carlo (QMC) simulations. As discussed in the previous section, these simulations consist of a thermalization phase in which no measurements are done, and a measurement phase in which measurements of Green's function and other observables (such as the two particle vertex) are accumulated. At a given cluster size N_c and temperature T, the QMC simulation is thermalized when the average expansion order $\langle k \rangle$ stabilizes – note that $\langle k \rangle \propto N_c^3\, T^{-3}$. Furthermore, in order to decorrelate the configurations between individual measurements, $\langle k \rangle$ updates have to be performed to the Green's function. Thus, the time needed to perform a QMC integration is characterized by the expansion order $\langle k \rangle$, the number or measurements needed to reach a desired accuracy, and the number of nodes that are available on the machine. We can empirically determine the thermalization time $\Delta_t(\langle k \rangle)$ and the time per measurement $\Delta_m(\langle k \rangle)$, i.e. time for $\langle k \rangle$ updates to the Green's function plus time for actual measurement. The parallel efficiency E for a given mean expansion order $\langle k \rangle$, number of nodes N_n, and necessary number of measurements N_m is then given by

$$E(N_n, N_m, \langle k \rangle) = \frac{1}{N_n} \frac{\Delta_t(\langle k \rangle) + N_m\, \Delta_m(\langle k \rangle)}{\Delta_t(\langle k \rangle) + \frac{N_m}{N_n}\, \Delta_m(\langle k \rangle)}. \qquad (11)$$

Consequently, the parallel efficiency remains high only as long as the number of measurements is large compared to the number of nodes used in the simulations. This is a typical Amdahl's Law behavior, where at large node counts the parallel efficiency drops at the moment that the parallel phase (measurements) becomes comparable to the serial phase (thermalization) of the simulation. A detailed comparison between this performance model and the real simulations at scale is given in Fig. 7 that will be discussed in the next section.

In Fig. 4 we show the time per measurement Δ_m when the algorithm is running on the CPU-only nodes of Monte Rosa and the hybrid CPU-GPU nodes of Todi. To be precise, the speed-up is the ratio between between Δ_m on an XE-6 node and an XK7 node.

For any given cluster size, we have to run a sequence of DCA$^{(+)}$ simulations, starting at high temperature and cooling down to temperatures below T_c, as we can only determine the critical temperature from the point where the leading eigenvalue of the two-particle vertex equals unity and the pair-field susceptibility diverges. From Fig. (4), it becomes clear that running the simulations on the hybrid CPU-GPU architectures is in almost all cases preferable. With the exception of small expansion order $\langle k \rangle$ where the N-matrices fit into the cache of the CPU, we obtain a considerable speed-up with the hybrid-multicore implementation, mainly due to the superior performance of DGEMM on the GPU compared to on a CPU. In particular when the mean expansion order is high ($\langle k \rangle \geq 1000$), i.e. at low temperature and large cluster sizes, we observe a very significant speed-up on the hybrid system. Since this is the region where the superconducting transition occurs and where a large number of measurements are needed to suppress the sign problem for large clusters, we have to focus our efforts to obtain a maximum speed-up and efficiency in this parameter regime. Hence, running the simulations on hybrid CPU-GPU nodes pays off from a time to solution perspective. This is even

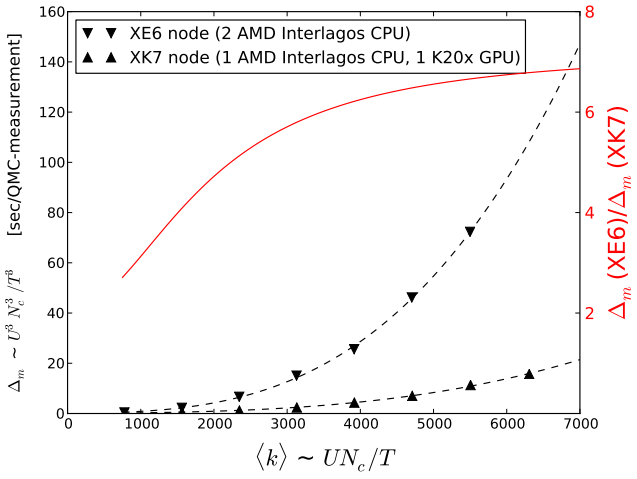
XE6 node (2 AMD Interlagos CPU)
XK7 node (1 AMD Interlagos CPU, 1 K20x GPU)
[sec/QMC-measurement]
$\Delta_m \sim U^3 N_c^3/T^3$
Δ_m (XE6)/Δ_m (XK7)
$\langle k \rangle \sim UN_c/T$

Figure 4: The seconds per measurement needed on a XE6 node (Monte Rosa) versus an XK7 node (Todi). The dotted lines are cubic fits to the data-points.

more the case from an energy to solution perspective. We find that despite the higher performance, the energy consumption of the XK7 cabinet is about equal to that of the XE6. A detailed comparison of the energy to solution, nor-

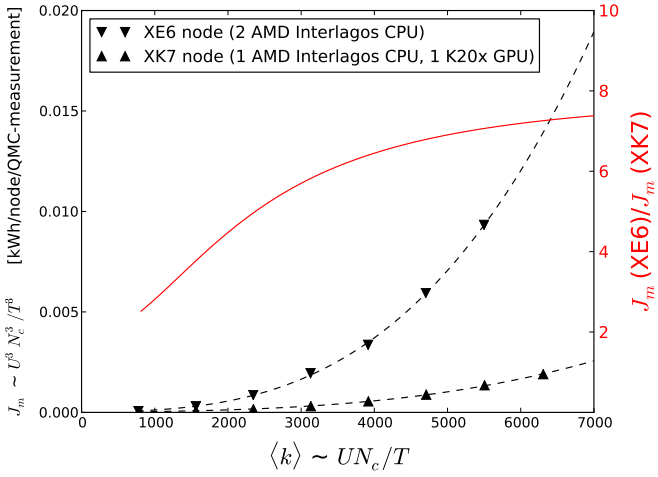
XE6 node (2 AMD Interlagos CPU)
XK7 node (1 AMD Interlagos CPU, 1 K20x GPU)
[kWh/node/QMC-measurement]
$J_m \sim U^3 N_c^3/T^3$
J_m (XE6)/J_m (XK7)
$\langle k \rangle \sim UN_c/T$

Figure 5: Energy (normalized by node and QMC measurement) needed to do a simulation on a XE6 (Monte Rosa) versus an XK7 node (Todi). The dotted lines are cubic fits to the data-points.

malized per node and measurement, is given in Fig. (5), where the results are determined from power measurements of the entire cabinet (thus including network and all cooling devices) taken at millisecond intervals and integrated over the entire simulation cycle. Using the GPU-based XK7 dramatically reduced the energy cost of the simulation. Thus, all simulations to produce the results plotted in figure 1 and those we will analyze in the next section, were done on the Cray XK7 supercomputers Todi and Titan.

6. PERFORMANCE RESULTS AT SCALE

The most important benefit of the DCA$^+$ algorithm is the improvement of the fermionic sign problem for larger clusters at lower temperatures. It allows us to converge the superconducting transition temperature T_c as a function of cluster size N_c. This was discussed in detail in section 3.2 and the results for T_c are plotted in figure 1. We will now discuss how the simulations to compute T_c were done on the large-scale computer systems with the implementation we have discussed in section 4. Since the implementation is portable to multi-core and hybrid CPU-GPU systems and the performance model of section 5 is generic, our present discussion should apply to all of high-end computing systems in operation today.

The reduction of the sign problem is of crucial importance to the time to solution. From the central limit theorem, we know that the standard deviation on an accumulated observable, i.e. the accuracy, is proportional to the inverse square root of the number of measurements. In QMC simulations, the fermionic algebra introduces contributions with a negative weight, often interpreted as a negative probability. To take these negative weights into account, we need to divide the accumulated observable with the average sign of the weights. This becomes problematic in the case of a small average sign. If we assume a standard deviation on our observable of the order of $\sigma_O \approx 1$, a reasonable assumption since all observables have a magnitude of the order of 1, we obtain that the number of measurements needed for a desired accuracy ϵ is

$$N_m = \left(\frac{\sigma_O}{\sigma_{\mathrm{QMC}}\,\epsilon}\right)^2 \qquad (12)$$

In order to obtain an estimate for the time-to-solution, we have to estimate two quantities: the time per measurement Δ_m and the fermionic sign. We know that the complexity of the time per measurement is cubic with inverse temperature $\Delta_m \propto T^{-3}$. In order to predict the fermionic sign, we fit a Fermi-Dirac like function $(1+\exp(-a(T-b))^{-1}$ to the measured average sign, which sets the positive parameters a and b (see inset of Fig. [6]). By multiplying the *time per measurement* with the desired number of measurements to obtain an accuracy of ϵ, we arrive at an empirical formula for the time-to-solution (TTS) for a desired accuracy ϵ at a given temperature T,

$$\mathrm{TTS(T},\epsilon) \propto \mathrm{T}^{-3}\left(\frac{\sigma_\mathrm{O}}{\epsilon}\right)^2\left(1+\mathrm{e}^{-\mathrm{a(T-b)}}\right)^2 \qquad (13)$$

There are two temperature regimes for the TTS. If $T > b$, the TTS will grow proportional T^{-3} since the fermionic sign will be close to 1. For $T < b$, the exponent will become more important and the TTS will grow exponentially. This can be clearly seen in figure 6, where the parameter b is approximately $\approx 0.2 - 0.25$ in the case of the DCA. Consequently, we can observe a kink in the log-log plot of the TTS for the DCA-simulations at those temperatures. In essence, the DCA$^+$ moves the transition between polynomial to exponential TTS to lower temperature due to its favorable sign and thus vastly improves the time to solution. Since we will never be able to run the DCA simulations at $U/t = 7$ and cluster sizes of $N_c = 12$ or larger with the machines we have available, we will have to estimate the TTS of the DCA from extrapolation down to T_c in figure 6. A comparison of the predicted TTS for the DCA$^+$ runs with actual TTS in actual $N_c = 28$ runs at $T \approx T_c$, shows that we have the correct order of magnitude. We estimate that *the DCA$^+$ algorithm improves the TTS of these runs by about 9 orders of magnitude compared to the state-of-the-art DCA runs*, using the same cluster solver and implementation in both methods.

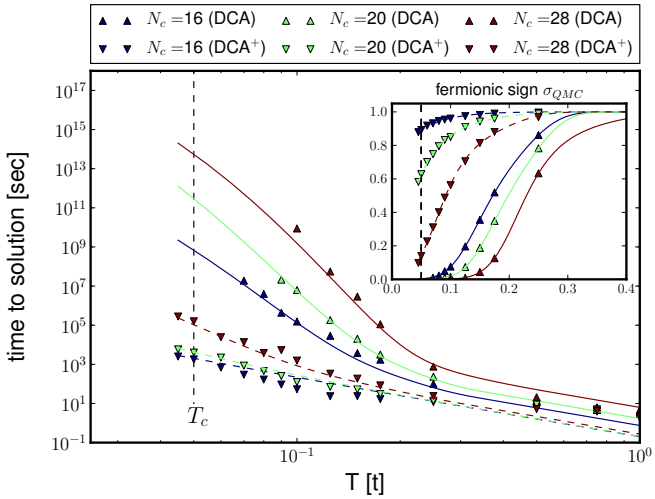

Figure 6: Comparison of the time to solution in the DCA and DCA$^+$ for various clusters at an accuracy of $\epsilon = 0.01$, at $U/t = 7$ and 10% doping. The solid and dotted lines are the estimated TTS according to Eq. (13). Inset: the evolution of the fermionic sign with temperature. The solid and dotted lines are the fits, used to generate the estimated TTS-curves.

Table 1: Effect of the number of measurements on the error of the self-energy $\Sigma_{\vec{K}}$ and T_c for $N_c = 28, U/t = 7$.

# measurements	accuracy on $\Sigma_{\vec{K}}$	T_c
500k	0.03	0.0484 ± 0.004
10M	0.0074	0.0518 ± 0.0014

The number of measurements necessary to reach a desired accuracy of $\epsilon = 0.01$ in the DCA$^+$ iterations at T_c, as estimated from eq. (12), is 10 million for $N_c = 28$ and $U/t = 7$, and 2.5 million for $N_c = 52$ and $U/t = 4$. These are the two largest, most time consuming runs we performed to produce the results plotted in figure 1. In table 1 we show for the measured ϵ_{QMC} is indeed in agreement with the desired value $\epsilon = 0.01$ that we used in the measurements, and that this value is indeed reasonable to reach an accuracy on T_c of a few percent. If only 500 thousand measurements were used, the error on T_c would be closer to 10%. The error on T_c shown in the table where determined from five independent runs with different seeds.

We have measured the parallel efficiency by running two sets of simulations, one with 500 thousand and one with 10 million measurements for $N_c = 28$ and $U/t = 7$ parameter-set. In figure 7 we compare the actual TTS of these simulations with the predictions from equation (11) for various number of nodes. We can conclude that the scaling limitations of the simulations are entirely driven by Amdahl's law and the relative time taken by the thermalization in the

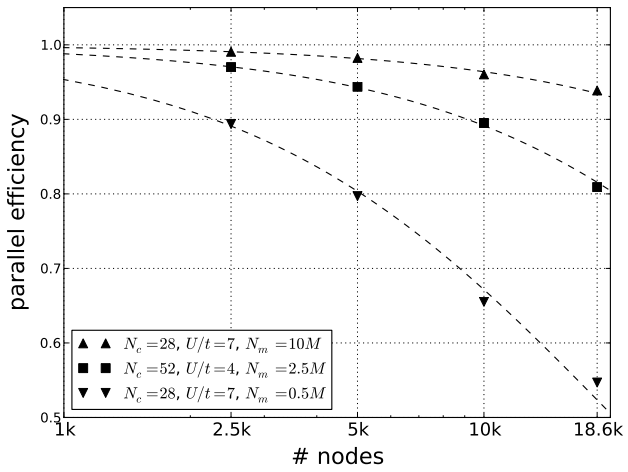

Figure 7: The parallel efficiency measured for different numbers of measurements in the strong ($U/t = 7$) and weak ($U/t = 4$) interaction limit. The dashed lines show the predictions of Eq. (11).

QMC cluster solver.

For the remainder of this analysis, we use the ($N_c = 28, U/t = 7$) and ($N_c = 52, U/t = 4$) simulations to study the strong- and weak-scaling behavior. As indicated above, in oder to reach a desired accuracy of $\epsilon = 0.01$, we have to run 10 million measurements for the former and 2.5 million for the latter. The corresponding strong scaling plot is shown in Fig 8. The measured parallel efficiency for the ($N_c = 28, U/t = 7$) run at 18,600 nodes is 0.94, in good agreement with estimates based on Eq. (11). A floating point performance plotted was determined with the method discussed in section 4. At 18,600 nodes, the floating point

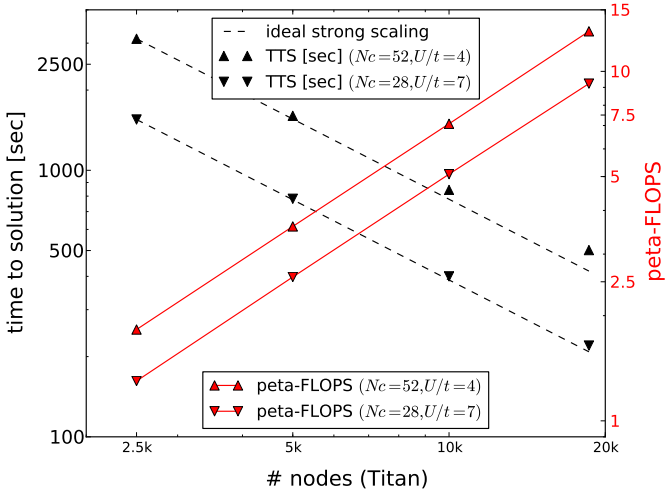

Figure 8: Strong scaling behavior for the largest clusters in the weak and strong interacting regime. The required 10 and 2.5 million measurements were used for $N_c = 28$ and $N_c = 52$, respectively, in order to reach a desired accuracy of $\epsilon = 0.01$ in the DCA$^+$ iterations

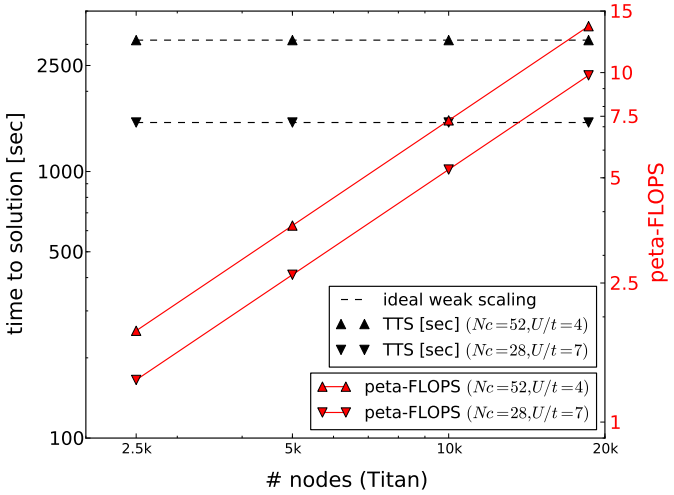

Figure 9: Weak scaling behavior for the largest clusters in the weak and strong interacting regime. The number of measurements are scaled proportionally with the number of nodes, with starting values at 2,500 node of 10 and 2.5 million for the $N_c = 28$ and $N_c = 52$ configurations, respectively.

performance is determined to be 0.94 for the 28 site cluster runs and 0.82 for the 52 site clusters. The relative difference in sustained performance is consistent with respective average expansion order $\langle k \rangle$ and corresponding matrix sizes used in the CT-AUX cluster solver.

For completeness, we also show weak scaling plots in figure 9. Here we start from the runs discussed above and same configurations used at 2,500 nodes in weak scaling experiments. The number of measurements are scaled proportionally with the number of nodes up to a maximum of 18,600. As expected, the time to solution remains constant. The sustained performance is 13.6 petaflops in the case of $N_c = 52$ and 9.7 petaflops when $N_c = 28$. The weak scaling plot indicate what we would have to do, if we ran with even larger clusters. In that case, we would have to increase the number of measurements even further to suppress the sign-problem. For example, if we were to run a 32-site cluster with $U/t = 7$, we estimate that we would need 100 million measurements, even more than the 10 million we used at 18,600 nodes in the strong scaling plot for $N_c = 28$. Furthermore, the 32-site runs will have a larger average expansion order and would thus take longer. We estimate that they would run for 100 minutes on 18,600 nodes on Titan.

The highest sustained performance in a full production run we have measured so far was 15.4 petaflops. This simulations ran on Titan (18,600 nodes) for 2250 seconds with a total energy consumption of 4300 kWh. It was recorded just prior to submission of the camera ready manuscript, while simulating the low doping region (5% hole-doping) of the Hubbard model in a 52-site cluster with $U/t = 7$. In this region of the phase-diagram, the lattice self-energy becomes strongly momentum dependent, thus requiring large clusters in order to have adequate momentum resolution. Had we done this run on a Cray XE6 with the same number of nodes but with multi-core processors only, we would have consumed 33,580 kWh and the time to solution would have

been more than six hours.

7. SUMMARY AND FUTURE WORK

We have presented a new algorithm to solve models of strongly correlated electron systems, and along with a scalable implementation that allows us to solve the 2D Hubbard model for high-temperature superconducting materials, such as the cuprates. The algorithm extends the state of the art Dynamical Cluster Approximation (DCA) with a continuous lattice self-energy in order to achieve better convergence with cluster size. The new method, which we call DCA$^+$, cures the cluster shape dependence problem of the DCA. The introduction of the continuous lattice self-energy removes artificial long-rage correlations and thereby significantly improves the fermionic sign problem of the underlying quantum Monte Carlo cluster solver in the entire temperature range of interest, and at large enough clusters to converge the computation of the superconducting transition temperature with cluster size.

A novel implementation of the state-of-the-art continuous time auxiliary field quantum Monte Carlo algorithm (CT-AUX) has been presented. It ports and runs with high efficiency on both multi-core and hybrid CPU-GPU systems. Using sub-matrix updates that improve data locality, the CT-AUX algorithm is optimized for CPU and GPU based systems. With this highly efficient implementation of the cluster solver, and running at full scale on the Cray XK7, Titan at ORNL, we are for the first time capable of computing the superconducting transition temperature T_c at 10% doping for realistic Coulomb repulsion $U/t = 7$ and at cluster sizes ($N_c = 28$) where T_c is seems converged at $T_c/t = 0.0518 \pm 0.0014$. For a smaller value of $U/t = 4$ we reach cluster sizes as large as 52 sites.

We have analyzed the performance of these full production runs that generate these new results. We find that the reduction of the QMC sign problem in the DCA$^+$ improves the time to solution for the $N_c = 28$ ($U/t = 7$) calculation by 9 orders of magnitude compared to state-of-the-art DCA simulations that use the same cluster solver. The behavior of the parallel efficiency in strong scaling experiments can be well understood in terms of Amdahl's law, where the scalability is entirely limited by the time spent in the thermalization phase of the CT-AUX cluster solver. In these strong-scaling experiments of the full production runs on 18,600 nodes of Titan, we measure a lower bound for the sustained performance to be 9.2 peta-flops in $N_c = 28$ case and 13.02 peta-flops when $N_c = 52$. Weak scaling experiments show, as expected for QMC, almost perfect scaling with a sustained performance on 18,600 nodes that is 9.8 peta-flops and 13.6 peta-flops for the $N_c = 28$ and $N_c = 52$ site cluster runs, respectively. In recent production runs that explore the phase-diagram of the Hubbard-model with a strong repulsion $U/t = 7$, we have observed a sustained performance of 15.4 petaflops on Titan in a 52-site cluster-run at 5% hole doping.

This DCA$^+$ algorithm opens entirely new avenues for simulations of strongly correlated systems. It will allow us to complete once and for all a full parametric study of the 2D Hubbard models, including the phase diagram as a function of doping, studying both the superconducting and the pseudo gap phases. Furthermore, we will be able to extend the model to multiple bands and include materials specific parameters derived from ab initio electronic structure calculations. This will allow us to compute T_c for specific cuprates and verify whether the approach allows us to reproduce the large variation of the transition temperature for different materials seen in experiment.

8. ACKNOWLEDGMENTS

We would like to thank Peter Messmer and Sadaf Alam for many enlightening discussion about implementation at scale, as well as Tiziano Belotti for providing us data on power consumption of the Cray XE7 and XE6 at CSCS. We are indebted to the help of Don Maxwell, Jack Wells, Jim Rogers, and the team at the Oak Ridge Leadership Computing facility (OLCF) for providing us timely access to Titan at full scale. This research was carried out with resources of the Swiss National Supercomputing Center (CSCS), OLCF, and the Center for Nanophase Materials Sciences (CNMS). OLCF and CNMS are located at Oak Ridge National Laboraroty and supported respectively by the Office of Science under Contract DE-AC05-00OR22725 and by the Scientific User Facilities Division, Office of Basic Energy Sciences, of the Department of Energy. CSCS is an autonomous unit of ETH Zurich.

9. REFERENCES

[1] G. Alvarez, M. S. Summers, D. E. Maxwell, M. Eisenbach, J. S. Meredith, J. M. Larkin, J. Levesque, T. A. Maier, P. R. C. Kent, E. F. D'Azevedo, and T. C. Schulthess. New algorithm to enable 400+ tflop/s sustained performance in simulations of disorder effects in high-T_c superconductors. In *Proceedings of the 2008 ACM/IEEE conference on Supercomputing*, SC '08, pages 61:1–61:10, Piscataway, NJ, USA, 2008. IEEE Press.

[2] O. Andersen, A. Liechtenstein, O. Jepsen, and F. Paulsen. LDA energy bands, low-energy hamiltonians, t', t'', $t_\perp(k)$ and $j_\perp$. *Journal of Physics and Chemistry of Solids*, 56(12):1573 – 1591, 1995. Proceedings of the Conference on Spectroscopies in Novel Superconductors.

[3] O. K. Andersen and T. Saha-Dasgupta. Muffin-tin orbitals of arbitrary order. *Phys. Rev. B*, 62:R16219–R16222, Dec 2000.

[4] V. I. Anisimov, D. E. Kondakov, A. V. Kozhevnikov, I. A. Nekrasov, Z. V. Pchelkina, J. W. Allen, S.-K. Mo, H.-D. Kim, P. Metcalf, S. Suga, A. Sekiyama, G. Keller, I. Leonov, X. Ren, and D. Vollhardt. Full orbital calculation scheme for materials with strongly correlated electrons. *Phys. Rev. B*, 71:125119, Mar 2005.

[5] F. Aryasetiawan, K. Karlsson, O. Jepsen, and U. Schönberger. Calculations of hubbard u from first-principles. *Phys. Rev. B*, 74:125106, Sep 2006.

[6] J. Bardeen, L. N. Cooper, and J. R. Schrieffer. Theory of superconductivity. *Phys. Rev.*, 108:1175–1204, Dec 1957.

[7] G. M. Eliashberg. Interactions between electrons and lattice vibrations in a superconductor. *Sov. Phys. JETP*, 11(3):696–702, 1960.

[8] A. Gonis. *Green functions for ordered and disordered systems*. Studies in mathematical physics. North-Holland, 1992.

[9] E. Gull, P. Staar, S. Fuchs, P. Nukala, M. S. Summers, T. Pruschke, T. C. Schulthess, and T. Maier. Submatrix updates for the continuous-time auxiliary-field algorithm. *Phys. Rev. B*, 83:075122, Feb 2011.

[10] E. Gull, P. Werner, O. Parcollet, and M. Troyer. Continuous-time auxiliary-field monte carlo for quantum impurity models. *Europhysics Letters*, 82(5):57003, 2008.

[11] M. H. Hettler, M. Mukherjee, M. Jarrell, and H. R. Krishnamurthy. Dynamical cluster approximation: Nonlocal dynamics of correlated electron systems. *Phys. Rev. B*, 61:12739–12756, May 2000.

[12] J. E. Hirsch and R. M. Fye. Monte carlo method for magnetic impurities in metals. *Phys. Rev. Lett.*, 56:2521–2524, Jun 1986.

[13] M. Jarrell, T. Maier, C. Huscroft, and S. Moukouri. Quantum monte carlo algorithm for nonlocal corrections to the dynamical mean-field approximation. *Phys. Rev. B*, 64:195130, Oct 2001.

[14] G. Kotliar, S. Y. Savrasov, K. Haule, V. S. Oudovenko, O. Parcollet, and C. A. Marianetti. Electronic structure calculations with dynamical mean-field theory. *Rev. Mod. Phys.*, 78:865–951, Aug 2006.

[15] A. Kozhevnikov, A. G. Eguiluz, and T. C. Schulthess. Toward first principles electronic structure simulations of excited states and strong correlations in nano- and materials science. In *Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '10, pages 1–10, Washington, DC, USA, 2010. IEEE Computer Society.

[16] E. Y. Loh, J. E. Gubernatis, R. T. Scalettar, S. R. White, D. J. Scalapino, and R. L. Sugar. Sign problem in the numerical simulation of many-electron systems. *Phys. Rev. B*, 41:9301–9307, May 1990.

[17] L. B. Lucy. An iterative technique for the rectification of observed distributions. *Astronomical Journal*, 79:745, June 1974.

[18] T. Maier, M. Jarrell, T. Pruschke, and M. H. Hettler. Quantum cluster theories. *Rev. Mod. Phys.*, 77:1027–1080, Oct 2005.

[19] T. A. Maier, M. Jarrell, T. C. Schulthess, P. R. C. Kent, and J. B. White. Systematic study of d-wave superconductivity in the 2d repulsive hubbard model. *Phys. Rev. Lett.*, 95:237001, Nov 2005.

[20] T. A. Maier, M. S. Jarrell, and D. J. Scalapino. Structure of the pairing interaction in the two-dimensional hubbard model. *Phys. Rev. Lett.*, 96:047005, Feb 2006.

[21] N. Marzari, A. A. Mostofi, J. R. Yates, I. Souza, and D. Vanderbilt. Maximally localized wannier functions: Theory and applications. *Rev. Mod. Phys.*, 84:1419–1475, Oct 2012.

[22] A. K. McMahan, J. F. Annett, and R. M. Martin. Cuprate parameters from numerical wannier functions. *Phys. Rev. B*, 42:6268–6282, Oct 1990.

[23] A. Migdal. Interaction between electrons and lattice vibrations in a normal metal. *Soviet Physics JETP*, 34(6):996–6, 1958.

[24] F. Mila. Parameters of a hubbard hamiltonian to describe superconducting cu oxides. *Phys. Rev. B*, 38:11358–11367, Dec 1988.

[25] M. R. Norman. The challenge of unconventional superconductivity. *Science*, 332(6026):196–200, 2011.

[26] P. K. V. V. Nukala, T. A. Maier, M. S. Summers, G. Alvarez, and T. C. Schulthess. Fast update algorithm for the quantum monte carlo simulation of the hubbard model. *Phys. Rev. B*, 80:195111, Nov 2009.

[27] S. Okamoto, A. J. Millis, H. Monien, and A. Fuhrmann. Fictive impurity models: An alternative formulation of the cluster dynamical mean-field method. *Phys. Rev. B*, 68:195121, Nov 2003.

[28] W. Richardson. Bayesian-based iterative method of image restoration. *Journal of the Optical Society of America*, 62(1):55–59, 1972.

[29] P. Staar, T. Maier, and T. C. Schulthess. Dynamical cluster approximation with continuous lattice self-energy. *arXiv:1304.3624v1*, Apr. 2013.

[30] P. Staar, T. A. Maier, and T. C. Schulthess. Efficient non-equidistant FFT approach to the measurement of single- and two-particle quantities in continuous time Quantum Monte Carlo methods. *Journal of Physics: Conference Series*, 402:012015, Dec. 2012.